Sentiment Analysis for Dynamic User Preference Inference in Spoken Dialogue Systems

Yves Vanrompay[1], Mario Cataldi[2], Marine Le Glouanec[1], Marie-Aude Aufaure[1], and Myriam Lamolle[2]

[1] MAS Laboratory, Ecole Centrale Paris
Grande Voie des Vignes,
F-92 295 Chatenay-Malabry Cedex, France
[2] Université Paris 8, Saint-Denis, France {yves.vanrompay,marine.le-glouanec, marie-aude.aufaure}@ecp.fr,{m.cataldi,m.lamolle}@iut.univ-paris8.fr

Abstract. Many current spoken dialogue systems for search are domain-specific and do not take into account the preferences of the user and his opinion about the proposed items. In order to provide a more personalized answer, tailored to the user needs, in this paper we propose a spoken dialogue system where user interests are expressed as scores in modular ontologies and his sentiment about the system propositions is considered. This approach allows for a dynamic and evolving representation of user interests. In fact, in order to improve the performance of the detection mechanism of users preferences, we propose a hybrid model which also makes use of a sentiment analysis module to detect the opinion of the user with respect to the proposition of the system. This allows the system to leverage the degree of user satisfaction and improve the overall recommendation mechanism being more precise about the expressed user interest. An evaluation on a representative set of dialogues is presented and highlights both the validity and the reliability of the proposed preference inference mechanism.

1 Introduction

The traditional goal of spoken dialogue system is to approach human performance in conversational interaction, specifically in terms of the interactional skills needed to do so. With this objective, as an attempt to enhance human-computer interaction, in the PARLANCE project[3], we build a system for interactive, *personalized*, *hyper-local* search for open domains such as restaurant search and tourist information. Current search engines work well only if the user has a single search goal and does not have multiple trade-offs to explore. For example, standard search works well if you want to know the phone number of a specific business but poorly if you are looking for a restaurant with several different search criteria of varying importance, e.g. food type versus location versus price etc. The latter requires the user to collaborate conversationally over several turns. In order to provide a personalized answer, tailored to the specific user, in

[3] https://sites.google.com/site/parlanceprojectofficial/

this work we focused on three levels: (1) user modelling in terms of preferences and interests inferred from past interactions with the system, (2) personalization of search approach and (3) the opinion of the user himself about the suggested products.

Personalization in the context of spoken dialogue system has slowly progressed compared to the field of non-natural language systems. Indeed, current spoken dialogue systems are mostly domain-specific, using rather static information from experts and knowledge bases. In PARLANCE, we choose to represent the dynamic domain knowledge through modular ontologies, where each ontology module represents a domain and can be dynamically loaded at run-time to meet the current needs of the user. In order to provide a semantic representation of the user model, the concepts and attributes in these ontology modules are annotated with scores representing the preferences and interests of the user. This allows us to learn the specificities of a user, and give responses that fit the user's profile. Additionally, user opinions can be detected where system recommendations are made. The sentiment analysis of the user answers permits to detect the degree of user satisfaction to systems replies and provide more tailored recommendations in the future.

This paper is organized as follows: Section 2 presents related work on preference systems. Section 3 motivates our research and provides an overview of the different components in the system. Section 4 introduces the representation of user interests while Section 5 describes how to detect and exploit user opinions about the recommended items. Then, we provide our experimental evaluation. Finally, section 7 concludes the paper.

2 Related Work

The automatic detection and categorization of sentiment expressed in natural language constitutes a remarkable research challenge, and has generated a nexus of techniques and approaches. Recently, [1] and [2] have conducted extensive surveys of the open challenges in this research area. Most of the works in this area focus on the categorization of *overall* sentiment, capturing a form of average polarity at the document level [3, 4].

A number of researchers have applied natural language processing (NLP) techniques to detect features in small chunks of text [5, 6]. Some methods make use of a lexical approach to focus on subjective words of the considered dialogue, namely adjectives and adverbs[7]. The study of linguistic complements, negations, syntactic dependencies, etc., can also be positively used to categorise the terms in the dialogue [8, 9].

Various API are available for sentiment analysis purposes. Most of them are built for study of specific environments, like social networks and/or blogs, and they try to focus on particular subjects, enterprises or events. Among all, SentiWordnet [10] analyses relationships like antonymy, hyponymy and/or hyperonymy to give a triplet value (objective, positive and negative) to each term of the considered phrase/document.

3 PARLANCE Architecture

This section starts with an example dialogue between the user and the system motivating the need for personalized interaction. Let us suppose that Paul is visiting San Francisco and he is looking for a good restaurant in the center of the city. An example dialogue between Paul (U) and the system (S) can be as follows:

1. U: I'm looking for a moderately priced restaurant near the centre of town.
2. S: I have found 2 Indian and 1 Chinese restaurant. Star India is a moderately priced restaurant in the centre of town. It has free Wifi.
3. U: Ok, what's the address?
4. S: The address is 3721 Geary Ave.
5. U: Oh, I don't want that one, it's hard to park on that street. Are there any other Indian restaurants?

In this conversation, not only information on user interests learned from past dialogues is exploited, but also new information on interests and preferences is learned. In item 2, the system has learned from previous conversations that the user is quite fond of Indian and Chinese restaurants. Therefore, it starts with proposing restaurants with these food types. The system also knows that the availability of Wifi is important to the user, so it proactively gives this information. In item 5, as the user asks for an alternative restaurant, user preferences are updated: the user does not like Geary Avenue.

Considering this scenario, the starting point for inferring user interests are abstract representations of the history of dialogues (dialogue act units) between the user and the system. A user model manager analyses the dialogue history of the user and derives interest scores associated to concepts, attribute types and attribute values in a weighted ontology module corresponding to a specific domain.

4 Evolving User Preferences as Weighted Modular Ontologies

User preferences regarding concepts and attributes are inferred from the user's dialogue history. All user and system utterances from past dialogues are saved in a so called dialogue act unit (DAU), which is the abstract representation of utterances in PARLANCE. For example, when the user asks for the price range of a restaurant, this is represented as the DAU *request(price)*. Interest scores are derived from logged traces of DAUs. We keep track of the positive and negative occurrences of attribute values and concrete instances. These frequencies allow us to rank the different elements. Based on this ranking it is decided which system response is best suited with regards to the user interests. If the user often queries for pricing information with given attribute value "cheap" in searching for a restaurant, the value of the *price* attribute will have a high frequency, and the system will proactively inform the user about the price in its answers, and lead the system to recommend restaurants from a cheaper price range.

Our mechanism for expressing user interests is integrated in the approach which represents information as (hierarchical) modular ontologies. Ontology modularization is defined as a way to structure ontologies, so that large domain ontologies will be the aggregation of self-contained, independent and reusable knowledge components (considered as Ontology module (OM)). An OM can be seen as an ontology fragment that has a meaning from the viewpoint of applications or users. Each ontology module corresponds to a particular domain and its size should be small for easy maintenance. Each OM is characterized by a basic concept, called the *pivotal concept*. A tourism ontology can contain several ontology modules like lodging, transportation and restaurant information. Taking the restaurant ontology module, this contains (amongst others) the restaurant concept with attribute types name, food type, dress code and location. To personalize the responses given to the user, our user model incorporates the interests and preferences of the user by assigning scores to elements in the appropriate ontology modules. These scores are updated according to what is being learned from the history of past dialogues, ensuring that the interests evolve as user preferences may change through time. The weights are useful in two different aspects. First, the scores are used to rank and recommend concrete instances that are of interest to users. Second, attribute value scores are used to generate a system response tailored to the user needs. For example, based on the interest scores, the system can decide to proactively inform the user on the food type of the restaurant, but not on the dress code.

4.1 Evolving update of the User Model

Calculation of scores happens offline based on all available dialogues in the dialogue history component. This means that the user model contains both the representation of user preferences (as weights in modular ontologies), as the mechanism to calculate the scores. To update scores based on recent dialogues of the user with the system, the user model manager aims to the recalculation of scores on a regular basis. The scores are thus **dynamic**, within the modular ontology structure which itself is relatively static. The scores of the attribute values are relative and sum up to 1. If an attribute type has m possible values, the initial score w_i for each value will be $\frac{1}{m}$. The score w_i is updated by counting how often attribute value $attval_i$ was selected and dividing this number by the total number of times a value for the corresponding attribute type was specified by the user. This means however that the past is as important as the present. In our context, user interests will typically evolve over time. So, if the dialogue history includes for instance the dialogues of the user during the last six months, it is reasonable to have more recent dialogues having relatively more influence on the user interest model than older ones. To this end, the scores for attribute values should be updated in such a way that recent dialogues have more relevance than older ones, which we do using an exponential smoothing method as follows: $w_i = \alpha \times x_j + (1-\alpha)w'_i$ where w'_i represents the old score and $x_j \in \{0,1\}$ is the value for the choice taken at moment j in the dialogue history. If $x_j = 1$ then $attval_i$ was specified by the user, if $x_j = 0$ it was not.

Using this method, the sum of all scores of the attribute values belonging to an attribute type remains 1, and the scores represent the relative importance of each attribute value. The *learning rate* $\alpha \in [0, 1]$ is a real number that controls how important recent observations are compared to older ones.

5 Mining User's Opinions wrt System Recommendations

This component is responsible for analyzing the positive, neutral, or negative opinions produced from user with respect to the propositions of the system. In order to tackle this issue, we make use of a novel feature-based polarity analysis technique[11], which combines statistical techniques with natural language processing. As in literature, we define a polarity as a real number that quantifies the user's positive, neutral, or negative opinion about a feature.

With this goal, for each dialog (intended as the complete set of system-user interaction), we model the user's sentiment with respect to the proposition of the user by estimating the degree of positivity/negativity with respect to the considered features. To extract such fine-grained sentiment information from raw text, we model each review as a set of sentences. A sentence is then formalized as a syntactic dependency graph, used to analyze the semantic and syntactic dependencies between its terms, and identify the terms referring to features. More formally, a sentence S can be formalize as an ordered vector of terms $S = \{w_0, w_1 ... w_m\}$, where the order represents the original position of each term within the sentence. The sentence s can be represented as a dependency graph G. The dependency graph is a labeled directed graph $G = (V, E, l)$, where V is the set of nodes representing the lexical elements w_i and E the set of edges (i.e. dependency relations) among the nodes.

The graph is obtained through a preliminary POS tagging phase, achieved by training a tagging model on the annotated corpus proposed by [12] and therefore by calculating the probability $p(t_j|w_i)$ of assigning a tag t_j to the term w_i using a maximum-likelihood estimation as in [13].

Subsequently, the dependency graphs are then utilized to detect the terms referring to a feature, which expresses some non-neutral opinion, including compound expressions, e.g. "the restaurant serves a very good pizza." In this phase, a SentiWordNet-like approach[10], which attributes polarity values to each WordNet synset, is used as a source of polarity values. In detail, using the synset graph proposed by WordNet, we calculate the polarities of each term by using a two-step algorithm. A first step is a semi-supervised learning step in which polarity values are assigned to two sets of *seed nodes*. This set consists of two subsets; one subset of "paradigmatically positive" synsets and another one consisting of "paradigmatically negative" synsets [14]. The polarities are then propagated automatically to other synsets of the WordNet graph by traversing selected semantic relations. This propagation is performed within the minimal radius that guarantees no conflicts among the relations, that is, until a node labeled as positive points to a node already linked to some negative seed, or vice-versa. In other words, we only propagate the polarities to the nodes that are univocally connected to a positive or a negative seed. Second, a random-walk step is executed on the whole WordNet graph starting from the seed nodes, and iteratively propagates the positive and negative polarity to all of the synsets. This approach preserves or inverts the polarity of each node based on the number of positive and negative relations that connect it to the seeds. The process ends when a convergence condition is reached. This condition is satisfied when all the nodes have maintained the same polarity sign (positive or negative) after two consecutive steps. Finally, the polarities of terms are aggregated into single values, each one referring to a specific feature.

6 Evaluation

The evaluation of our approach consists in two different analysis; from one side, capturing the evolution of the user preference scores in time, i.e. with respect to the size of the dialogue history or the number of interactions of the user with the system, and from the other side, study the user feedback expressed in the dialog with the system.

6.1 Analysing the evolution of the user preference scores

By using the fading factor α in our interest score update formula, we make sure that more recent dialogues provide more information on interests than the older ones, leading to an evolution of interests. To obtain real spoken dialogues, we used Amazon Mechanical Turk, which is a tool for crowd-sourcing where users can call a toll-free number, solve tasks assigned to them, and earn money. First, a number of well defined tasks, expressed in natural language, were constructed. As an example, we ask a user to find an Italian restaurant in the center of town. As a check for task success, the user has to give in the phone number of the restaurant he has found. This means the test users have every incentive to succeed, since they are only paid in case of task success. We varied the content of the tasks as to reflect changing user interests over time. The basic task was in each case to find a restaurant with variations in the attribute types *food*, *area* and *price*. The experiments are based on 60 dialogues, and we set α to be 0.1. In analyzing the evolution of the interest scores, we plot the maximum score P across attribute values for each attribute type as a function of the number of dialogues considered, as can be seen in Figure 1. Indeed, the score P serves as a basic metric for showing how outspoken the user interests are at a certain moment in time. A low P signifies that the user changes his interest rather quick, while a high P means that in recent dialogues he has shown a rather consistent and stable choice behavior.

Figure 1 shows the evolution of the values of P over the dialogues for different attribute types. It can be noticed that there is a correlation between the "peaks" in the graph for *food* and *area*, reflecting that when the user is in a stable period with respect to his interests, this holds for some attribute types. The evolution of the graph allows us to identify periods in which the user very dynamically changes his interests, and periods in which his choices remain merely stable. Also, by looking at the relative values of P for the different attribute types at a given moment, it is possible to make a ranking of those types where the user does not change his behavior a lot, compared to the ones that are more fluctuating.

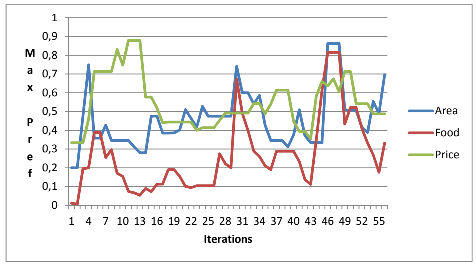

Fig. 1. Evolution of user interests in function of dialogue history

6.2 Dissecting user opinion with respect to System recommendations

In order to assess the proposed approach to sentiment analysis, we analysed the detected sentiments in the considered corpus of dialogs. The aim of this analysis is to study, from one side, how the users interact with the system and, on the second side, analyse the retrieved opinion with respect to the recommended items. For this, we considered three features: food type, area and price. Figure 2 shows the obtained polarities.

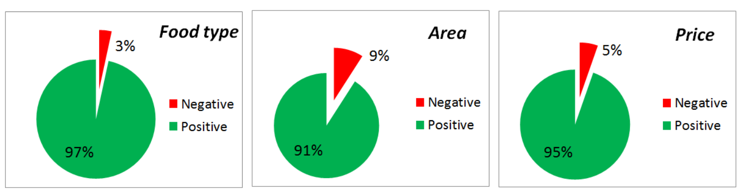

Fig. 2. Statistics about the performed feature-based polarity on user dialogs.

These results highlight, for all the considered features, that the user express more likely positive opinions rather than negative ones. In a sense, the users accept, in great majority of the cases (up to 97% of the cases regarding the food type) the recommendations of the system, explicitly proving the goodness of the whole recommender system and the user preference model. In fact, the previously collected preferences lead the system to recommend an item that perfectly match the real user preferences.

Notice that this feedback value can be positively use to dynamically tune the *learning rate* α (explained in Section 4.1). In fact, a negative opinion can suggest a change of the user preference wrt the attended one. Thus, we can dynamically set the learning rate, which express how important recent observations can be compared to older ones, proportionally to the negative sentiment expressed by the user in order to reflect a sort of user preference change. Following this idea, the more negative the sentiment expressed by the user, the higher the learning rate which will reflect the necessity of the system to quickly update the user preferences.

7 Conclusion

In this paper we described an approach for modeling the dynamics of user preferences in a spoken dialogue system for searching items of interest. A fading method allows for the interests to keep track of the evolution of user behavior. We then leverage the opinion of the user about the suggested items to improve the recommender system and tune the learning rate of the system. We evaluated our approach on a set of real dialogues and showed it can provide useful insights into changing interests. The ontology-based representation of interests lets us tailor recommendation of items to the recent preferences the user has exhibited, for each search domain involved.

References

1. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval **2**(1-2) (2008) 1–135
2. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies **5**(1) (2012) 1–167
3. Beineke, P., Hastie, T., Manning, C., Vaithyanathan, S.: Exploring Sentiment Summarization. In: Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications. (2004) 1–4
4. Hiroshi, K., Tetsuya, N., Hideo, W.: Deeper Sentiment Analysis Using Machine Translation Technology. In: Proceedings of COLING '04. (2004) 1–7
5. Holz, F., Teresniak, S.: Towards Automatic Detection and Tracking of Topic Change. In: Computational Linguistics and Intelligent Text Processing. Volume 6008. (2010) 327–339
6. Missen, M., Boughanem, M., Cabanac, G.: Opinion Mining: Reviewed from Word to Document Level. (2012) 1–19
7. Turney, P.D.: Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the ACL '02. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 417–424
8. Benamara, F., Chardon, B., Mathieu, Y.Y., Popescu, V.: Towards context-based subjectivity analysis. In: IJCNLP. (2011) 1180–1188
9. Wilson, T., Wiebe, J., Hwa, R.: Just how mad are you? finding strong and weak opinion clauses. In: Proceedings of the AAAI'04, AAAI Press (2004) 761–767
10. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Proceedings of LREC'10. (2010) 2200–2204
11. Cataldi, M., Ballatore, A., Tiddi, I., Aufaure, M.A.: Good location, terrible food: detecting feature sentiment in user-generated reviews. Social Netw. Analys. Mining **3**(4) (2013) 1149–1163
12. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics **19**(2) (1993) 313–330
13. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics. ACL '03 (2003) 423–430
14. Turney, P.D., Littman, M.L.: Measuring Praise and Criticism: Inference of Semantic Orientation from Association. ACM Trans. Inf. Syst. **21**(4) (October 2003) 315–346